Readme

自动化报告生成说明文档

本代码库提供了一套自动化解决方案，旨在高效地从 **PPT模板** 中提取结构化信息，并结合 **用户需求** 智能生成报告的配置。其核心在于利用 **大语言模型（LLM）** 的强大能力，将复杂的自然语言需求转化为精确的结构化数据，进而生成可执行的数据库查询和报告配置。

工作原理

整个流程可分为两大核心步骤：

1. 提取PPT模板内的相关信息

此步骤主要由 `pptx_parser.py` 和 `text_utils.py` 协同完成，旨在将PPT文件中的视觉元素及其布局信息提取出来，并进行结构化处理。

- **PPT解析（`pptx_parser.py`）**：
  - 使用 `python-pptx` 库遍历PPT中的幻灯片和所有形状（Shape）
  - 通过 `_get_shape_type` 方法，将形状识别为 **图表**（`chart-bar`，`chart-line` 等）、**表格**（`table`）或 **文本框**（`text`）。
  - 记录每个形状的 **布局信息**（位置和尺寸），并计算其中心点坐标。
  - 通过计算中心点距离，将文本框（特别是标题）与相邻的图表或表格进行 **智能匹配**(这个地方暂时先这么做，后面再根据情况看是否需要修改)。
- **文本信息解析（`text_utils.py`）**：
  - `extract_details_from_title` 函数，它使用 **正则表达式** 对PPT标题文本进行解析。
  - 例如，从标题 `"2021-2023年北京市怀柔区供应与成交趋势"` 中，准确提取出 **时间范围**（2021-2023年）、**地名**（北京市、怀柔区）和 **报告意图**（供应与成交）。
  - 这些结构化信息（`data_range`、`block`、`intent`）为后续大模型的处理和SQL生成提供了基础。

整个过程最终会生成一个 YAML 格式的模板结构，包含了幻灯片的尺寸、标题、分析文本以及每个内容元素（图表或表格）的类型和布局信息。

2. 对需求进行解析并提取关键信息（使用大模型）

此步骤是整个流程的“大脑”，由 `sql_generator.py` 中的 `SqlGenerator` 类负责。它利用 **大语言模型**（如 DeepSeek-Chat）来理解用户的自然语言需求，并将其转化为机器可执行的指令。

- **需求输入：** 用户的需求，例如 `"基于该模板，请生成2021-2023年北京市怀柔区怀柔区板块的详细报告"`。
- **Prompt 模板：** `SqlGenerator` 定义了两个关键的 Prompt 模板：
  - **SQL 生成模板：** 指示大模型扮演“SQL专家”角色，并提供数据库表结构和问答示例，使其能直接生成正确的 **SQL 语句**。
  - **数据源 JSON 生成模板：** 指示大模型扮演“数据提取专家”，并要求它将用户的需求（如城市、区域、时间）提取为一个结构化的 **JSON 对象**，用于配置报告的数据源。
- **LLM 交互：**
  - `generate_sql(user_question)` 方法调用大模型接口，返回一个可执行的 SQL 查询。
  - `generate_datasource_json(user_question)` 方法则返回一个包含关键信息的 JSON 对象（Python字典）。

最终，`yaml_processor.py` 会将从PPT中解析出的模板结构、大模型生成的数据源信息和SQL查询，整合到一个完整的、可用于后续数据填充和图表生成的 YAML 配置文件中。

代码结构

- `main.py`：主入口文件，负责加载任务、初始化生成器，并使用多线程并发处理所有报告任务。
- `file_utils.py`：处理文件I/O，如查找CSV文件、读取任务列表以及加载/保存YAML文件。
- `pptx_parser.py`：负责解析PPTX文件，提取幻灯片结构、形状信息（类型、位置、尺寸）。
- `text_utils.py`：辅助PPTX解析，主要通过正则表达式从标题中提取时间、地点等信息。
- `sql_generator.py`：核心模块，封装了与大语言模型（DeepSeek-Chat）的交互，用于生成SQL查询和结构化的JSON数据。
- `yaml_processor.py`：将所有步骤（PPT解析、大模型生成）的结果整合，生成最终的YAML配置文件并保存。

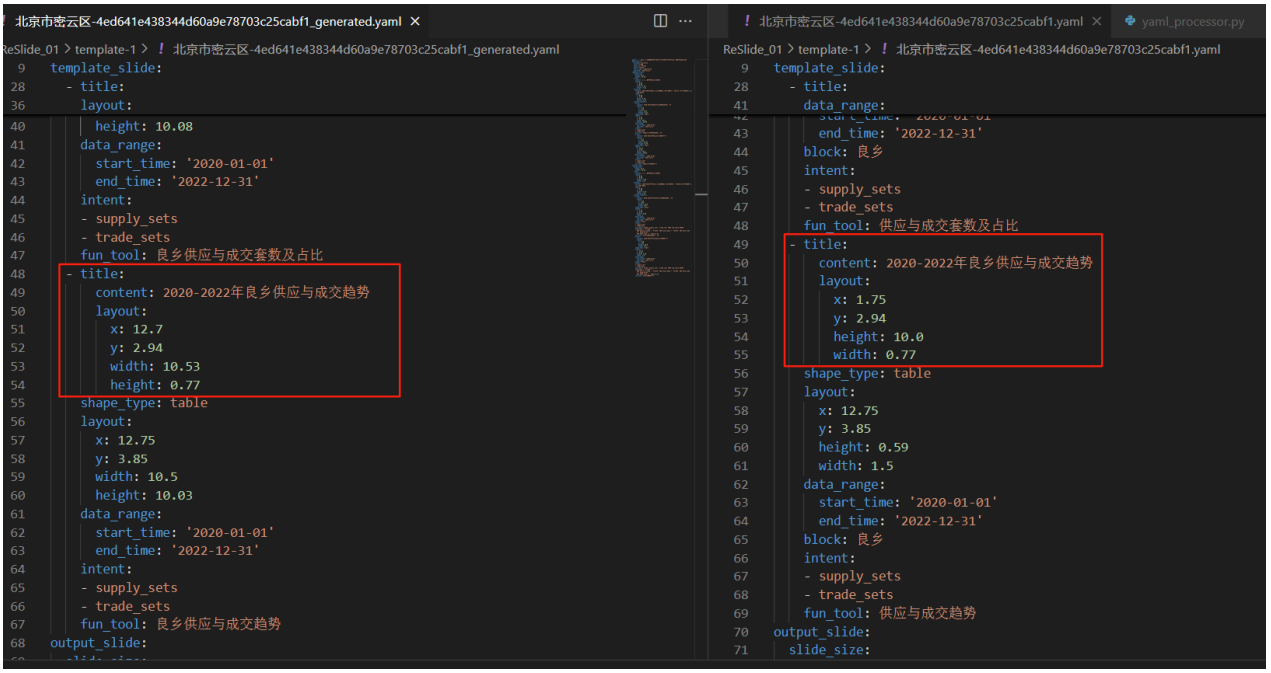
使用方法

1. **环境准备**：安装所需的Python库，如 `python-pptx`，`pyyaml`，`langchain-deepseek` 等。
2. **配置API Key**：在项目根目录下创建一个 `.env` 文件，并配置 DeepSeek API Key：`DEEPSEEK_API_KEY='your_key'`。
3. **准备任务**：在指定的目录结构（`ReSlide_*/template-*/temp/`）下放置一个 `filename_to_label.csv` 文件，其中包含PPTX模板路径、用户需求和对应的真值YAML文件路径。

4. **运行程序**：执行 `python main.py`，程序将自动查找所有任务并并发生成新的YAML配置。

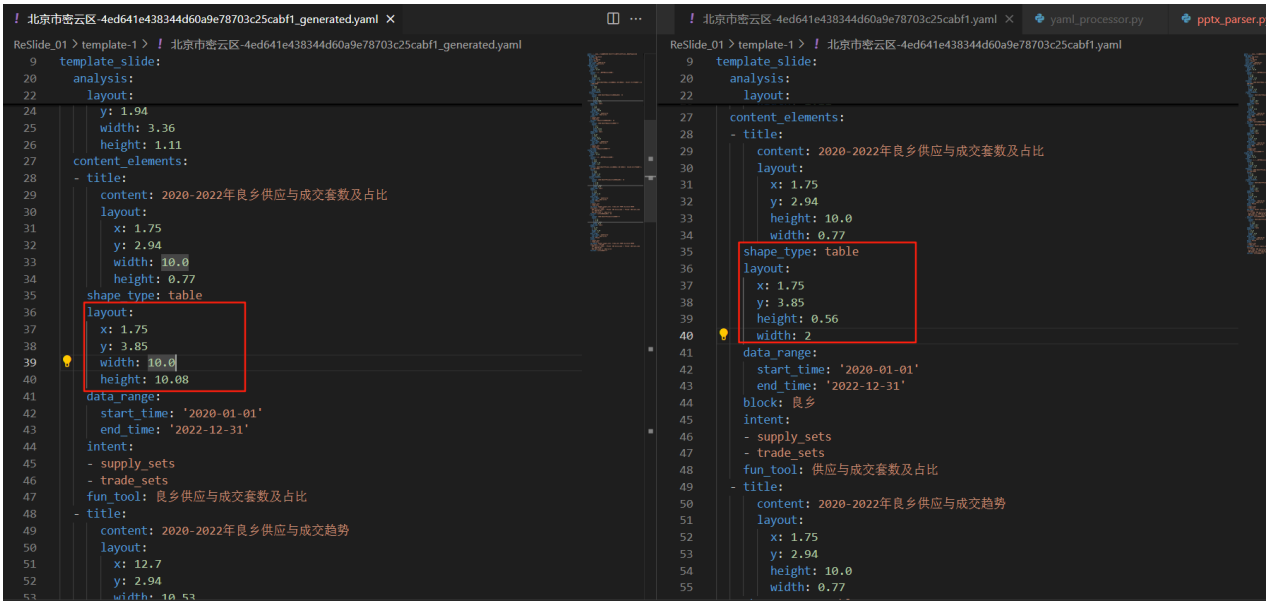
说明

1. output_slide的内容，除了'sql_query'键外，暂时放的是原本yaml的内容(因为没有数据)，预留了相关接口(`YamlProcessor._generate_output_slide`)供后面修改
2. 设计了一个小的`evaluation.py`脚本来评测sql生成效果，使用的是sqlite，如果使用mysql的话需要对代码做一些修改，感觉效果还好
3. 查看了一下ppt模板匹配出来的结果，有一些坐标略微对不上，但感觉还好



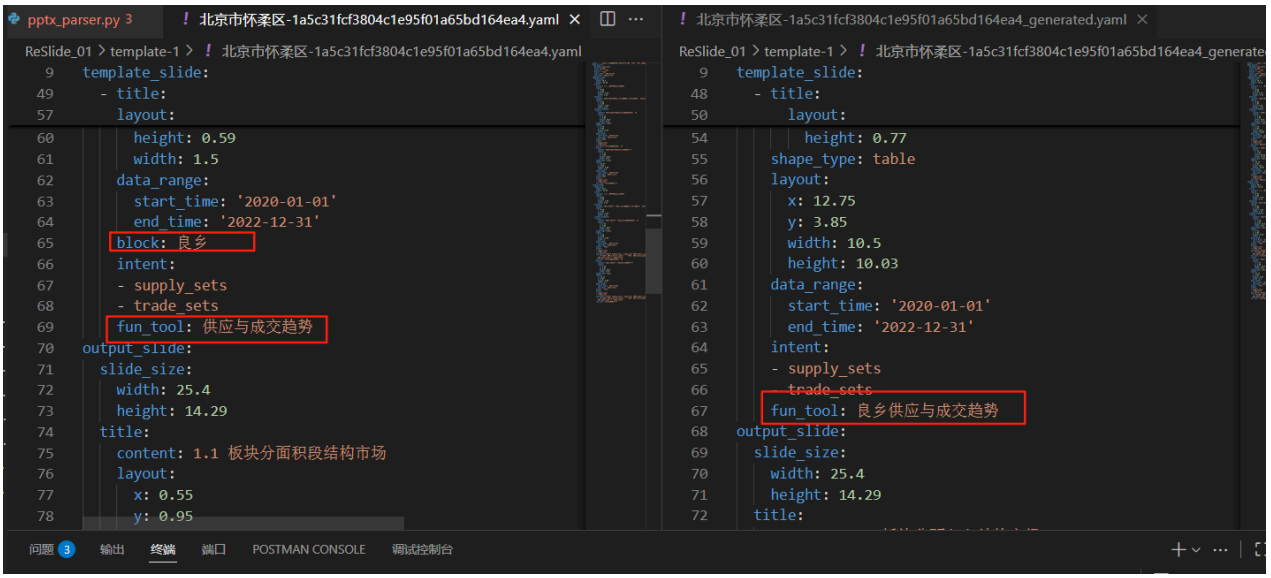
(左边是提取的，右边是ground truth)

4. Reslide_01/template-1的template_slide，表格宽度和高度好像跟真实情况有点对不上？



左边是提取模板得到的表格宽高，右边是原本给的yaml ground truth的表格宽高

5. 'block'键和'fun_tool'键暂时还没有处理好，因为不是很好用正则拆开，如果用LLM来专门处理这个，感觉也没有太大必要，看后面需求决定？


pptx_parser.py 3
北京市怀柔区-1a5c31fcf3804c1e95f01a65bd164ea4.yaml
ReSlide_01 > template-1 > 北京市怀柔区-1a5c31fcf3804c1e95f01a65bd164ea4.yaml
template_slide:
- title:
layout:
height: 0.59
width: 1.5
data_range:
start_time: '2020-01-01'
end_time: '2022-12-31'
block: 良乡
intent:
- supply_sets
- trade_sets
fun_tool: 供应与成交趋势
output_slide:
slide_size:
width: 25.4
height: 14.29
title:
content: 1.1 板块分面积段结构市场
layout:
x: 0.55
y: 0.95
北京市怀柔区-1a5c31fcf3804c1e95f01a65bd164ea4_generated.yaml
ReSlide_01 > template-1 > 北京市怀柔区-1a5c31fcf3804c1e95f01a65bd164ea4_generate
template_slide:
- title:
layout:
height: 0.77
shape_type: table
layout:
x: 12.75
y: 3.85
width: 10.5
height: 10.03
data_range:
start_time: '2020-01-01'
end_time: '2022-12-31'
intent:
- supply_sets
- trade_sets
fun_tool: 良乡供应与成交趋势
output_slide:
slide_size:
width: 25.4
height: 14.29
title:
问题 3
输出
终端
端口
POSTMAN CONSOLE
调试控制台